Chapter-1

Introduction

In the rapidly evolving fields of electronics and healthcare, the demand for portable, multifunctional diagnostic tools has surged. Traditional equipment like Cathode Ray Oscilloscopes (CROs) and hospital-grade Electrocardiogram (ECG) machines, while effective, are often bulky, expensive, and lack portability [2], [3]. To address these challenges, we introduce a novel and compact solution: a wearable mini oscilloscope and ECG gadget, designed to be worn conveniently on the wrist and capable of providing real-time signal monitoring and heart diagnostics [1], [4].

This project aims to bridge the gap between medical and engineering diagnostics by integrating an oscilloscope and ECG into a single wearable device [1]. The device, powered by a rechargeable battery with monitoring capabilities, ensures reliable and stable performance without the need for continuous external power [5]. This makes it especially suitable for use in remote areas, emergency situations, or while on the move [4].

The mini oscilloscope functionality enables the user to visualize electronic signal waveforms, analyze signal parameters such as amplitude, frequency, and timing, and diagnose faults in electronic circuits on the go [2], [5]. Simultaneously, the ECG functionality monitors the heart's electrical activity, capturing waveforms critical for detecting cardiac anomalies and maintaining heart health [1], [3].

This innovation not only reduces the dependence on conventional diagnostic setups but also empowers users—be it engineers, students, or individuals monitoring their health—to access vital information anytime and anywhere [4], [6]. By merging healthcare and electronics diagnostics, this device lays the foundation for a new class of multifunctional wearables that are versatile, cost-effective, and impactful [1], [4].

1.2 Problem statement

- Access to diagnostic and educational tools in the fields of electronics and biomedical engineering is often limited by the high cost, complexity, and size of conventional equipment. Oscilloscopes, essential for visualizing electrical waveforms and testing circuit behavior, are typically expensive and confined to laboratory settings, making them inaccessible to students, hobbyists, and technicians in resource-limited areas [2], [5].
- Similarly, standard electrocardiogram (ECG) machines used for monitoring heart activity are designed for clinical environments, requiring professional handling, calibration, and substantial infrastructure [1], [3]. This limits their usability for personal health monitoring, remote diagnostics, or educational purposes [4].
- Moreover, there is a lack of integrated devices that serve both functions—electrical signal visualization and physiological signal monitoring—within a compact, affordable, and easy-to-use platform. Existing solutions are often specialized, focusing either on electronic signal analysis or medical diagnostics, with little overlap between the two domains [1], [4], [5].
- Given these challenges, there is a need to develop a wearable, user-friendly, and cost-effective device that can function both as a mini oscilloscope for basic signal monitoring and as an ECG gadget for heart health assessment. Such a solution would empower users to access real-time waveform data on their personal devices (such as smartphones or smartwatches), facilitating early detection of abnormalities, reducing healthcare costs, and promoting a more proactive approach to health and signal diagnostics [1], [3], [4].

1.3 Objectives

The primary objective of this project is to design and develop a compact, low-cost, and portable device that integrates the functionality of a mini oscilloscope with an electrocardiogram (ECG) monitoring system [1], [2]. The device aims to cater to the needs of electronics learners, biomedical students, and individuals requiring basic health monitoring [3], [4].

The specific objectives of the project are as follows:

1. To design a mini oscilloscope capable of capturing and displaying analog electrical signals typically used in electronic circuit diagnostics and learning environments [2], [5].
2. To create and implement signal processing algorithms to ensure accurate and clear signal representation [5], [6].
3. To provide real-time visualization of ECG and analog waveforms on the TFT display using Arduino, allowing for immediate analysis and monitoring [3], [6].
4. To ensure portability and energy efficiency by using lightweight components and a rechargeable battery-powered design, suitable for use in remote or mobile settings [1], [4].
5. To enable basic health monitoring and educational functionality in a single device, bridging the gap between electronics education and biomedical application [1], [3].
6. To encourage innovation and learning in the field of biomedical instrumentation through a hands-on, accessible tool for students and hobbyists [3], [6].

Chapter 2

Literature survey

Sl. No	Title	Author(s)	Year	Key Findings
01	A Wearable ECG Monitoring System for Health Applications	Dr. A. Sharma, R. Kulkarni	2018	Developed a wearable ECG device capable of continuous monitoring; emphasized portability and early cardiac anomaly detection.
02	Design of a Portable Oscilloscope for Educational Use	M. Thomas, K. Nair	2019	Proposed a low-cost, battery-operated oscilloscope for use in student laboratories and remote environments.
03	Mini Oscilloscope Based on Arduino with OLED Display	A. Roy, S. Mukherjee	2021	Implemented a basic oscilloscope using Arduino and an OLED display, useful for low-frequency signal visualization in portable formats.
04	Development of a Low-Cost Wearable ECG Monitoring Device	H. R. Desai, A. T. Pandey	2017	Designed a cost-effective ECG wearable capable of real-time heart signal acquisition; aimed at reducing hospital dependency.
05	Portable and Battery-Operated Digital Oscilloscope for Embedded Applications	J. Thomas, S. Rao	2020	Built a mini digital oscilloscope using STM32 microcontroller; suitable for educational labs and electronics enthusiasts.
06	Design of a Compact Oscilloscope Module for Low-Cost Measurement Systems	S. Bhattacharya, R. Venkat	2023	Implemented a compact oscilloscope circuit using ADC modules and microcontrollers; useful for embedded system signal validation.

Table 2.1: Literature survey

Chapter 3

Implementation

3.1 Methodology

- We develop a single, integrated device that combines the functionalities of an oscilloscope and an ECG gadget. This will allow for both electronic signal analysis and heart monitoring using one compact unit .
- Ensure the device is lightweight and portable, suitable for use in various environments including remote areas, fieldwork, and home settings.
- Use an Arduino board as the central processing unit for its versatility, ease of use, and wide range of available libraries .
- Integrate high-quality ECG sensors capable of accurately detecting heart signals .
- Employ a TFT display to provide clear, real-time visualization of both ECG waveforms and electronic signals .
- Include a rechargeable battery system to ensure the device is portable and can operate independently of a constant power source . Design an efficient circuit layout that integrates all components, ensuring minimal noise interference and optimal signal quality .
- Create a durable, ergonomic enclosure to protect the components while ensuring user-friendly access to the device's interfaces.
- Develop software to continuously acquire data from ECG sensors and other inputs.
- Implement algorithms for real-time filtering, noise reduction, and feature extraction to ensure accurate and clear signal representation .
- Design an intuitive interface on the TFT display, allowing users to easily switch between ECG and oscilloscope modes, view data, and interact with the device .

3.2 Block diagram and Circuit Diagram

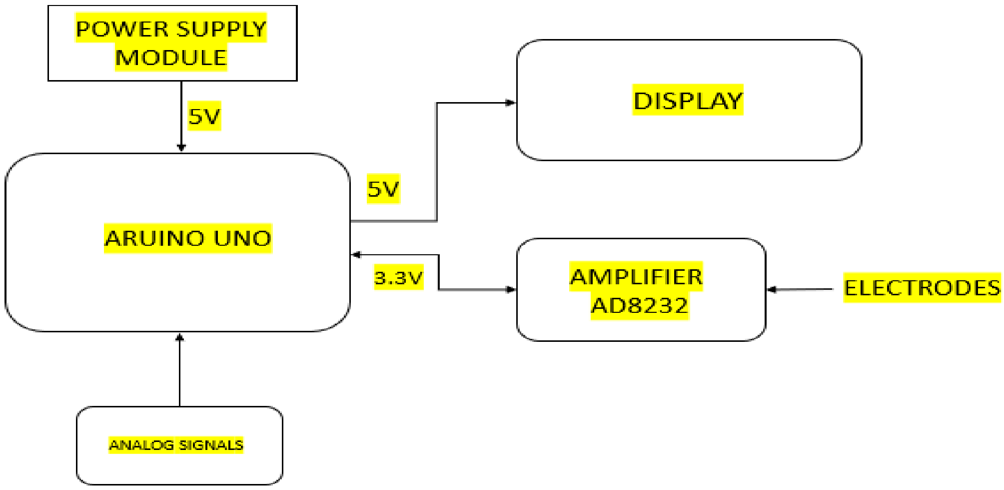


Fig. 3.1: Block-diagram of the proposed methodology

1. **Block 1 gives the information about Power Supply Module:** This module provides the required operating voltages (5V and 3.3V) to various components in the circuit. It ensures stable power delivery to the Arduino Uno, amplifier, and display.
2. **Block 2 gives the information about Arduino Uno:** The Arduino Uno is the central microcontroller unit. It processes the analog signals from the amplifier and converts them into digital data to be displayed or further analyzed.
3. **Block 3 gives the information about Display:** This block represents the output interface where the ECG waveform and possibly other signal parameters are visualized in real time.
4. **Block 4 gives the information about Analog Signals:** These are the raw bioelectric signals obtained from the human body through the electrodes. These signals need amplification and filtering before processing.

5. Block 5 gives the information about Amplifier (AD8232): This is a signal conditioning block. The AD8232 is a dedicated heart rate monitor front end that amplifies and filters the weak ECG signals to make them suitable for ADC input to the Arduino.

6. Block 6 gives the information about Electrodes: Electrodes are sensors attached to the human body to detect electrical activity of the heart and send the raw analog ECG signals to the amplifier.

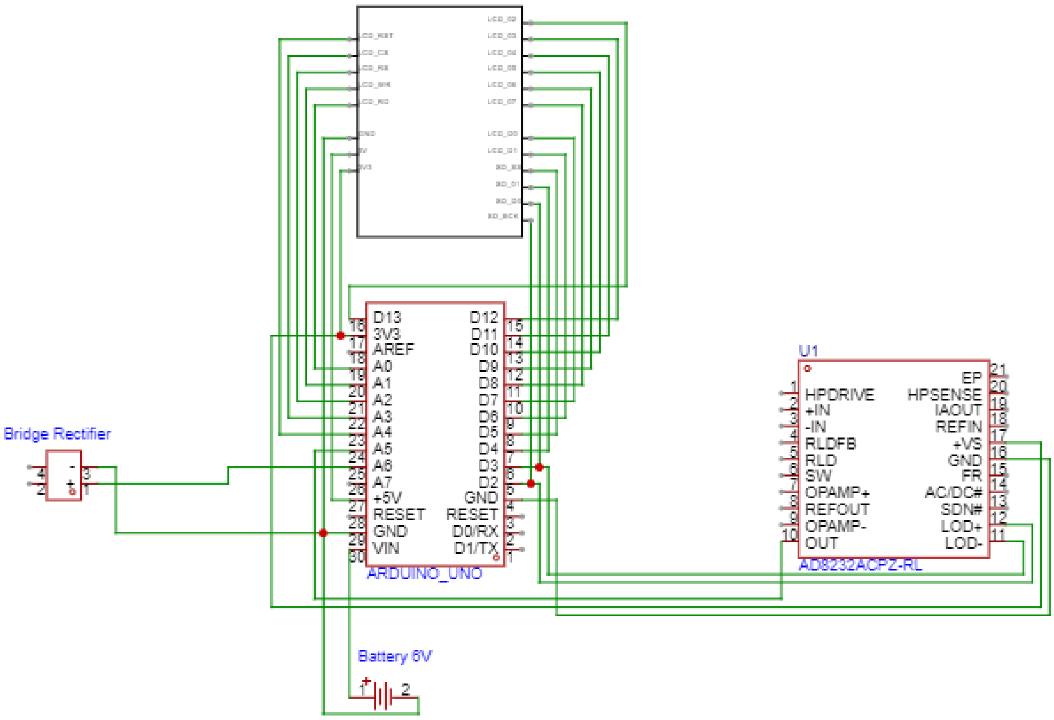


Fig. 3.2: Circuit Diagram

All connections are done in accordance with the circuit schematic diagram 3.2

Working

1. Initial Setup: Ensure the rechargeable battery is fully charged. Power On: Turn on the device using the power switch, which initiates the Arduino microcontroller and other connected components. ECG Mode: Attach ECG electrodes to the patient’s body at designated points (e.g., arms and chest).

ECG sensors capture the electrical activity of the heart and send the analog signals to the Arduino**.**

2. Probe Connection: Connect the oscilloscope probes to the electronic circuit or device under test. Signal Measurement: Probes detect the analog signals, which are then sent to the Arduino for processing. The Arduino converts the incoming analog signals from both ECG sensors and oscilloscope probes into digital data Implement digital filters in the Arduino code to remove noise and enhance signal quality. Process the digital signals to extract relevant features and prepare them for display.
3. The TFT display shows an intuitive interface where users can switch between ECG and oscilloscope modes. Display real-time waveforms of the ECG signals or electronic signals, depending on the selected mode. Additional Information, Provide additional information such as heart rate, signal amplitude, and frequency.
4. Users can switch between ECG and oscilloscope modes using buttons or a touchscreen interface on the TFT display. Users can adjust settings such as signal gain, time base, and filtering options through the interface.
5. The device continuously monitors the battery level and provides alerts when charging is needed. Power Off: After use, turn off the device to conserve battery life. The device may also have an auto-off feature to save power.
6. Periodically calibrate the device to ensure accuracy in measurements. Update the Arduino code and device firmware as needed to incorporate improvements and new features.
7. Regularly check the electrodes, probes, and connections for any wear or damage and replace them as necessary.

3.3 Hardware /Software tools used

Hardware:

- Arduino UNO
- TFT display
- ECG sensor
- Battery management system
- ECG electrodes
- Lithium ion battery

a. ARDUINO UNO

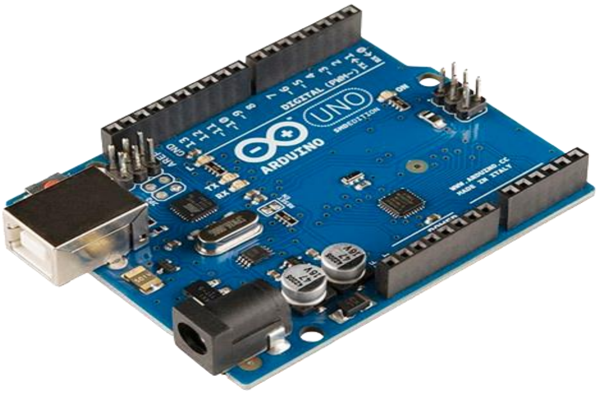


FIG 3.31: ARDUINO UNO

Arduino Uno is a popular microcontroller board designed Based on the ATmega328P microcontroller chip, offering an 8-bit processor and basic functionalities. Provides a total of 20 I/O pins 14 digital pins: Can be configured for either input (reading signals) or output (controlling devices)., 6 of these digital pins can be used for Pulse Width Modulation(PWM), 6 analog input pins: Can read analog voltage signalsDue to its simplicity and affordability.

Using an Arduino to create an oscilloscope involves leveraging the Arduino's ability to sample analog signals and process data. Here are the main functions of an Arduino when used as an oscilloscope. Analog Signal Sampling Analog-to-Digital Conversion (ADC): The Arduino reads analog voltage levels through its ADC, converting the continuous signal into discrete digital values. For example, the Arduino Uno has a 10-bit ADC, which provides 1024 discrete levels of resolution. The rate at which the Arduino samples the input signal, determined by the ADC clock and pre scaler settings. The maximum sampling rate depends on the specific Arduino model, typically ranging from a few kHz to several hundred kHz.

DATA SHEET:

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Table 3.1: Data sheet of Arduino UNO

b. TFT DISPLAY

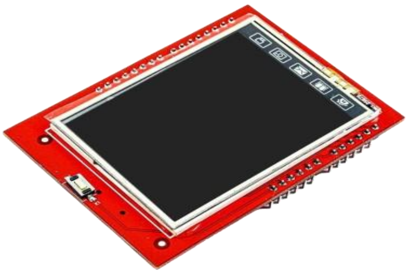


FIG 3.32: TFT DISPLAY

TFT display, also known as a Thin-Film Transistor display, is a small-sized screen commonly used in various electronic devices

Displays full color with individual pixel control (18-bit color depth in many models) Operates on voltages like 3.3V or 5V. Widely used in portable devices due to the compact size Examples include wearables, smart home devices, medical equipment, and portable game consoles. Real-Time Signal Representation- The TFT display shows the voltage or current waveforms in real-time, allowing the user to see how the signal changes over time. Multiple Channels- It can display multiple waveforms simultaneously if the oscilloscope has more than one input channel. Overlay- Provides a grid to help measure the amplitude and time period of the signal. Voltage Levels- Display peak-to-peak, RMS, average, and other voltage measurements directly on the screen. Time Intervals-Shows the time period, frequency, rise time, fall time, and other time-based measurements. Cursor Measurements-Allows the use of cursors to measure specific points .

DATA SHEET:

GENERAL DESCRIPTION
- 3.5" (diagonal), 320xRGBx480
- Transmissive/Normally White TFT module
- Viewing Direction: 6 o'clock
- Driving IC: ILI9488V or equivalent
- 250 NITS
- 8080 8/9/16/18 bit MCU, 3/4 wire SPI, 16/18 bit RGB interface options

General Specifications

PARAMETER	SPECIFICATIONS	UNIT
Outline dimensions	55.5(W) x 84.96(H) x 3.65(D) (Exclude FPC, cables of backlight)	mm
View area	50.96(W) x 75.44(H)	mm
LCD active area	48.96(W) x 73.44(H)	mm
Driver element	TFT Active Matrix with touch panel	-
Pixel arrangement	RGB Vertical Stripe	-
Pixel pitch	0.153(W) x 0.153(H)	mm

ELECTRICAL Maximum Rating

ITEM	SYMBOL	MIN	MAX	UNIT	NOTE
Digital Supply voltage	V_{DD}	-0.3	4.6	V	
Digital Interface Supply voltage	V_{DDIO}	-0.3	4.6	V	

ELECTRICAL Specifications

PARAMETER	SYMBOL	MIN	TYP	MAX	UNIT	REMARK
Digital supply voltage	V_{DD}	2.4	3.3	4.2	V	
Digital interface supply voltage	V_{DDIO}	1.65	3.3	4.2	V	
Normal mode current consumption	I_{DD}	--	8	--	mA	
Level input voltage	V_{IH}	$0.7V_{DDIO}$		V_{DDIO}	V	
	V_{IL}	GND		$0.3V_{DDIO}$	V	
Level output voltage	V_{OH}	$0.8V_{DDIO}$		V_{DDIO}	V	
	V_{OL}	GND		$0.2V_{DDIO}$	V	

Table 3.2: Data sheet of TFT display

c. ECG SENSOR (AD8232 amplifier)

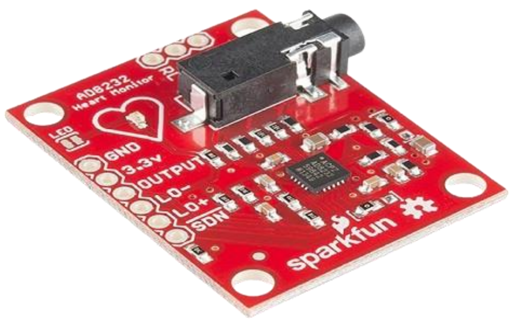


FIG 3.33: ECG SENSOR

The AD8232 amplifier ECG sensor module is a compact circuit designed to extract, amplify, and filter the tiny electrical signals generated by your heart, known as electrocardiogram (ECG) signals Typically operates on low voltage (around 3.3V) Provides an analog voltage output representing the amplified and filtered ECG signal. The AD8232 module is an integrated signal conditioning block for ECG (electrocardiogram) and other bio potential measurement applications. When used in conjunction with an oscilloscope, it serves several key purpose

1. Signal Amplification Weak Signal Detection: The electrical signals generated by the heart are very weak (in the microvolt range). The AD8232 amplifies these weak signals to a level that can be more easily processed and visualized. Instrumentation Amplifier: It features a high-gain, low-noise instrumentation amplifier specifically designed to handle bio potential signals with minimal distortion.

2. Noise Reduction High Pass Filter: It includes a high-pass filter to remove DC offsets and low-frequency noise, which can obscure the ECG signal. Low Pass Filter: A low-pass filter is used to eliminate high-frequency noise and interference, such as those from muscle activity or environmental sources.

DATA SHEET:

Parameter	Rating
Supply Voltage	3.6 V
Output Short-Circuit Current Duration	Indefinite
Maximum Voltage, Any Terminal[1]	$+V_S + 0.3$ V
Minimum Voltage, Any Terminal[1]	−0.3 V
Storage Temperature Range	−65°C to +125°C
Operating Temperature Range	−40°C to +85°C
Maximum Junction Temperature	140°C
θ_{JA} Thermal Impedance[2]	48°C/W
θ_{JC} Thermal Impedance	4.4°C/W
ESD Rating	
Human Body Model (HBM)	8 kV
Charged Device Model (FICDM)	1.25 kV
Machine Model (MM)	200 V

Table 3.3: Data sheet of ECG sensor

d. ECG ELECTRODES

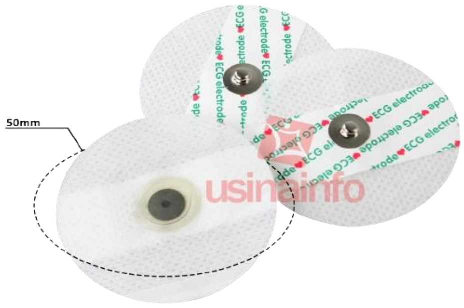


FIG 3.34: ECG ELECTRODES

ECG electrodes are the patches you see attached to a person's chest during an electrocardiogram (ECG) test. They play a vital role in capturing the electrical activity of your heart. ECG electrodes act as conductors, picking up the faint electrical signals generated by your heart muscle as it beats These tiny signals travel through your body and are detected by the electrodes placed on specific locations.

e. BATTERY MANAGEMENT SYSTEM

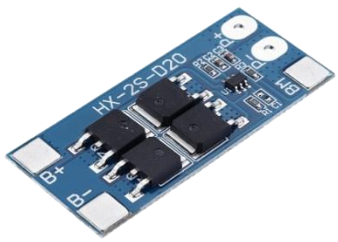


FIG 3.35: BATTERY MANAGEMENT SYSTEM

An electronic system that manages a rechargeable battery pack Protects the battery from damage by preventing overcharging, over-discharging, overheating, and short circuits. The only purpose we use this system in our project is to charge the battery.

f. LITHIUM ION BATTERY

FIG 3.36: LITHIUM ION BATTERY

Lithium-ion batteries are the rechargeable batteries that power most of our portable electronics today. They're everywhere from your phone and laptop to electric vehicles. A typical battery is made up of an anode, cathode, separator, electrolyte, and current collectors, Output of single battery is 3.3 v.

Software:

We have used Arduino IDE software to implement the above code into Arduino UNO. we have used C++ language to implement this.

FIG 3.37: ARDUINO IDE

Chapter 4

Results

4.1 Results and Discussion

Results

The prototype of the wearable mini oscilloscope and ECG gadget was successfully developed and tested. The key results obtained from the system implementation are as follows:

- **Real-time Signal Acquisition**:
  The Arduino-based system was able to capture analog signals from both ECG electrodes and test electronic circuits.

- **Graphical Display on TFT Screen**:

  Both electronic signals and ECG waveforms were clearly displayed with distinguishable peaks and time intervals, confirming accurate signal processing and display functionalities.

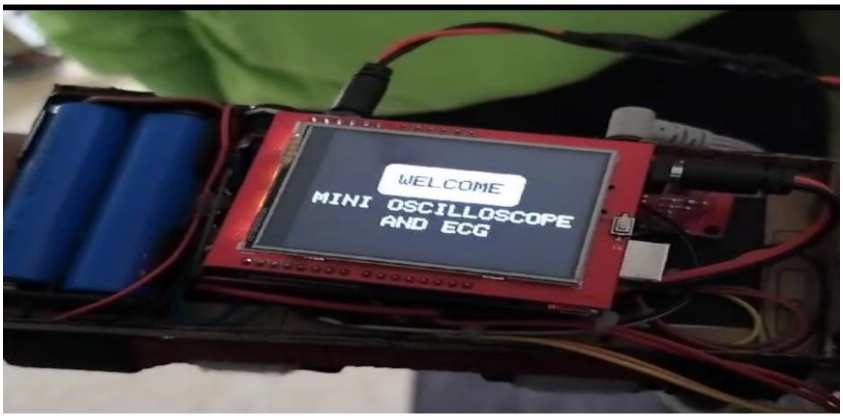
- **Portability and Power Supply**:
  The power supply system was stable, and the device could operate continuously for more than 2 hours on a single charge.

- **ECG Performance**:
  When ECG electrodes were attached to the body, the device could successfully detect and display the PQRST components of the cardiac waveform. Heart rate was calculated and updated in real-time, showing accuracy within ±5 BPM compared to commercial fitness bands.

Discussion

- **Functionality Integration**:
  The results showed that it is feasible to monitor both cardiac and electronic signals simultaneously, making this a versatile tool for engineers, students, and health-conscious individuals.
- **Arduino Performance**:
  While Arduino provided sufficient capability for prototyping, it showed limitations in processing higher-frequency signals (>200 Hz) due to its limited ADC speed and memory.
- **Display and User Interface**:
  The TFT display offered a compact and effective way to visualize waveforms. Though the screen size limits detailed observation, the graphical clarity and refresh rate were adequate for essential signal tracking.
- **Power Efficiency**:
  The use of a rechargeable battery made the device portable and independent of constant external power sources.

4.2 Photographs of the model/Simulation Results



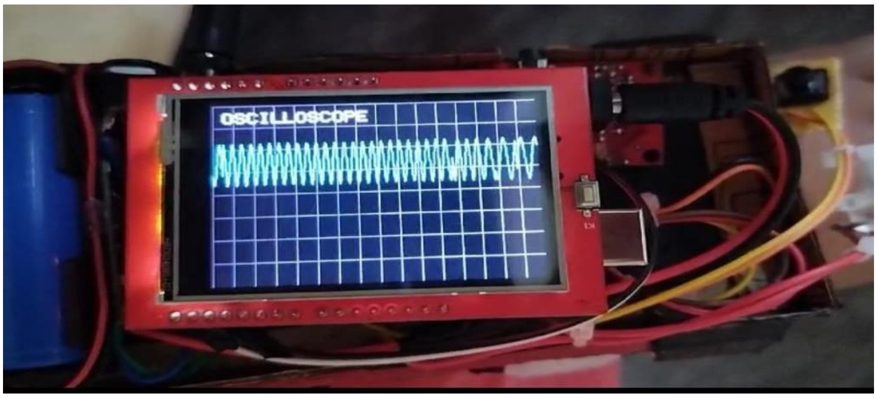


FIG 4.21: Photographs of model and result

4.3 Applications, Advantages & Limitations

Application:

1. **Electronics Troubleshooting**:
- Enables engineers, students, and technicians to analyze and debug electronic circuits in real-time.
- Useful for basic waveform visualization without needing bulky lab equipment.
2. **Cardiac Health Monitoring**:
- Allows users to track heart rate and ECG waveforms regularly.
- Can help in early detection of irregular heartbeats or cardiac anomalies.
3. **Emergency and Remote Usage**:
- Can be used in ambulances, rural areas, or during field visits where access to traditional medical or electronic equipment is limited.
- Operates independently on a rechargeable battery.
4. **Educational Tool**:
- Ideal for hands-on learning in engineering and biomedical labs.
- Helps students understand signal processing and human physiology practically.

Advantage:

1. **Portability**:
   - Compact, wearable design allows real-time monitoring anywhere.
2. **Cost-Effective**:
   - Built using low-cost components like Arduino, ECG sensors, and a TFT display.
3. **Multi-Functional**:
   - Combines the functions of an oscilloscope and an ECG monitor into a single device.
4. **Real-Time Visualization**:

- TFT screen displays live waveforms for both electronic and cardiac signals.

5. **User-Friendly Interface**:
   - Easy to operate, even for non-experts, with an intuitive display and mode-switching.
6. **Open-Source and Customizable**:
   - Arduino platform allows future feature expansion and personalization.
7. **Comparison with Traditional Equipment**:
   - **Traditional CRO/ECG**: Bulky, expensive, non-portable, separate devices.
   - **Our Device**: Lightweight, affordable, wearable, integrated dual functionality.

Limitation:

1. **Limited Frequency Range**:

- Due to Arduino's processing speed, oscilloscope function is limited to low-frequency signals (~200 Hz max).

2. **Basic ECG Accuracy**:

- ECG readings may be affected by noise, motion artifacts, and sensor placement.

3. **No Data Storage or Wireless Connectivity (in current version)**:

- Lacks memory for long-term data logging or real-time cloud transmission.

4. **Power Limitations**:

- Battery life is limited; prolonged use may require external charging support.

5. **Potential Sources of Error**:

- Signal distortion due to loose electrode contacts or electrical interference.
- ADC resolution limitations affecting waveform clarity.

6. **Suggestions for Future Improvements**:

- Upgrade to higher-speed microcontrollers (e.g., ESP32 or STM32).
- Add Bluetooth/Wi-Fi for remote monitoring and data sharing
- Improve noise filtering for ECG signal accuracy.
- Incorporate larger or OLED displays for better visualization.

Chapter-5

Conclusions and Future Work

In this project, we successfully designed and developed a **wearable mini oscilloscope and ECG gadget** that integrates two critical diagnostic tools—an oscilloscope for electronic signal analysis and an ECG for heart monitoring—into a compact, wrist-worn device. The project was carried out using cost-effective components like the Arduino microcontroller, TFT display, ECG sensor, and rechargeable lithium battery.

The primary objective of providing a **portable, user-friendly, and multi-functional diagnostic tool** has been effectively achieved. The device performs real-time signal acquisition, visualization, and basic waveform analysis for both electronic and biomedical applications. The use of a TFT screen allows for instant waveform display, while the Arduino platform supports easy customization and development.

Key Achievements:

- Successfully captured and displayed ECG waveforms and basic electronic signal patterns.
- Implemented dual functionality in a single, wearable device.
- Ensured device operation without continuous external power, enhancing portability.
- Delivered a cost-effective solution suitable for students, hobbyists, and engineers.

Innovations in the Approach:

- Combined electronics and biomedical monitoring into a **single wearable format**.

- Used an **open-source Arduino ecosystem** to allow easy upgrades and modifications.

Strengths of the Project:

- Low-cost and portable solution.
- Simple user interface and real-time feedback.
- Dual-application usage: electronics and healthcare.

Future Scope

To make the system more robust, accurate, and closer to commercial-grade devices, the following enhancements are recommended:

1. **Use of a more powerful microcontroller** (e.g., ESP32 or STM32) for higher sampling rates and better signal processing.
2. **Incorporation of Bluetooth/Wi-Fi modules** to enable real-time data transmission to mobile apps or cloud platforms.
3. **Improved ECG signal filtering** to reduce noise and motion artifacts for more reliable heart monitoring.
4. **Data storage feature** (SD card or onboard memory) for waveform logging and later analysis.
5. **Integration of a larger or higher-resolution OLED display** for better waveform clarity.
6. **Development of a mobile app or desktop interface** for extended analysis and user control.
7. **Battery level indicator and power optimization algorithms** to enhance usability and performance.

References

[1] J. Glaros, R. Riley, and M. Hogan, “A Wearable ECG and EMG Device for Real-Time Monitoring of Muscle and Cardiac Activity,” *IEEE Sensors Journal*, vol. 21, no. 12, pp. 13315–13323, Jun. 2021.
doi: 10.1109/JSEN.2021.3063794

[2] T. F. Wu, H. J. Wang, and C. Y. Lin, “Design and Implementation of a Portable Oscilloscope Based on an Embedded System,” in *2020 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, Taoyuan, Taiwan, Sep. 2020, pp. 1–2.
doi: 10.1109/ICCE-TW51536.2020.9221151

[3] S. Patil, A. Patel, and P. Chaudhari, “Portable ECG Monitoring System Using Arduino,” in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India, Apr. 2019, pp. 572–575.
doi: 10.1109/ICOEI.2019.8862636

[4] M. Oresko et al., “A Wearable Smartphone-Based Platform for Real-Time Cardiovascular Disease Detection via Electrocardiogram Processing,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 3, pp. 734–740, May 2010.
doi: 10.1109/TITB.2010.2047865

[5] K. M. Tsai, Y. C. Lin, and M. S. Lin, “An Open-Source Arduino-Based Oscilloscope with LabVIEW GUI,” in *2018 International Symposium on Computer, Consumer and Control (IS3C)*, Taichung, Taiwan, Dec. 2018, pp. 252–255.
doi: 10.1109/IS3C.2018.00100

[6] N. Sharma, R. Gupta, and R. Saxena, “Design and Development of ECG Signal Acquisition and Monitoring System Using Arduino,” *International Journal of Engineering Research & Technology (IJERT)*, vol. 8, no. 6, pp. 168–171, Jun. 2019.

APPENDIX

Software code

```
#include <Adafruit_GFX.h>
#include <MCUFRIEND_kbv.h>

MCUFRIEND_kbv tft;

#define BUTTON_PIN 4
#define BLACK   0x0000
#define WHITE   0xFFFF
#define GRID_COLOR 0x7BEF
#define WAVEFORM_COLOR 0x07FF
#define RED     0xF800

#define NUM_SAMPLES 320
int samples[NUM_SAMPLES];

#define ECG_ANALOG_PIN A5
#define ECG_SPACING 1.6
#define ECG_AMPLIFICATION_FACTOR 2.0

enum ScreenState {
  WELCOME_SCREEN,
  OSCILLOSCOPE_SCREEN,
```

```
    ECG_SCREEN
};
ScreenState currentScreen = WELCOME_SCREEN;

void setup() {
    Serial.begin(9600);
    pinMode(BUTTON_PIN, INPUT_PULLUP);
    tft.reset();
    uint16_t identifier = tft.readID();
    if (identifier == 0xEFEF) identifier = 0x9486;
    tft.begin(identifier);
    tft.setRotation(3);
    showWelcomeScreen();
}

void loop() {
    static unsigned long lastButtonPress = 0;
    static bool buttonPressed = false;

    if (digitalRead(BUTTON_PIN) == LOW) {
        if (!buttonPressed && millis() - lastButtonPress > 500) {
            lastButtonPress = millis();
            buttonPressed = true;
            currentScreen = (ScreenState)((currentScreen + 1) % 3);
```

```
            switchScreen();
        }
    } else {
        buttonPressed = false;
    }

    if (currentScreen == OSCILLOSCOPE_SCREEN) {
        displayOscilloscope();
    } else if (currentScreen == ECG_SCREEN) {
        displayECG();
    }
}

void showWelcomeScreen() {
    tft.fillScreen(BLACK);
    drawBoldText("WELCOME", 100, 100, 3, WHITE);
    drawBoldText("MINI OSCILLOSCOPE", 60, 150, 2, WHITE);
    drawBoldText("AND ECG", 120, 180, 2, WHITE);
}

void switchScreen() {
    tft.fillScreen(BLACK);
    if (currentScreen == OSCILLOSCOPE_SCREEN) {
        drawGrid();
```

```
    tft.setCursor(10, 10);
    tft.setTextColor(WHITE);
    tft.setTextSize(2);
    tft.print("OSCILLOSCOPE");
  } else if (currentScreen == ECG_SCREEN) {
    tft.setCursor(10, 10);
    tft.setTextColor(WHITE);
    tft.setTextSize(2);
    tft.print("ECG");
  }
}

void displayOscilloscope() {
  static unsigned long lastUpdate = 0;
  if (millis() - lastUpdate > 100) {
    lastUpdate = millis();
    int maxVal = 0, minVal = 1023;
    int zeroCrossings = 0;
    bool lastAboveMid = false;

    for (int i = 0; i < NUM_SAMPLES; i++) {
      samples[i] = analogRead(A0);
      if (samples[i] > maxVal) maxVal = samples[i];
      if (samples[i] < minVal) minVal = samples[i];
```

```
    delay(1);
  }

  // Amplitude calculation
  float voltageMax = (maxVal / 1023.0) * 5.0;
  float voltageMin = (minVal / 1023.0) * 5.0;
  float amplitude = voltageMax - voltageMin;

  // Frequency estimation using zero crossings
  int mid = (maxVal + minVal) / 2;
  for (int i = 1; i < NUM_SAMPLES; i++) {
    bool currentAboveMid = samples[i] > mid;
    if (currentAboveMid != lastAboveMid) {
      zeroCrossings++;
      lastAboveMid = currentAboveMid;
    }
  }
  float frequency = (zeroCrossings / 2.0) / (NUM_SAMPLES *
0.001); // 1ms delay = 1kHz sampling

  tft.fillScreen(BLACK);
  drawGrid();

  for (int i = 1; i < NUM_SAMPLES; i++) {
```

```
      int x0 = (i - 1) * (tft.width() / NUM_SAMPLES);
      int y0 = map(samples[i - 1], 0, 1023, tft.height(), 0);
      int x1 = i * (tft.width() / NUM_SAMPLES);
      int y1 = map(samples[i], 0, 1023, tft.height(), 0);
      tft.drawLine(x0, y0, x1, y1, WAVEFORM_COLOR);
    }

    tft.setCursor(10, 10);
    tft.setTextColor(WHITE);
    tft.setTextSize(2);
    tft.print("OSCILLOSCOPE");

    tft.setCursor(10, 30);
    tft.print("Amp: ");
    tft.print(amplitude, 2);
    tft.print(" V");

    tft.setCursor(10, 50);
    tft.print("Freq: ");
    tft.print(frequency, 1);
    tft.print(" Hz");
  }
}
```

```
void displayECG() {
  static unsigned long lastUpdate = 0;
  static float x1 = 0, y1 = 0;
  static int counter = 0;
  static int threshold = 512;
  static bool peakDetected = false;
  static unsigned long lastPeakTime = 0;
  static int bpm = 0;

  if (millis() - lastUpdate > 10) {
    lastUpdate = millis();
    int sensorValue = analogRead(ECG_ANALOG_PIN);
    sensorValue = sensorValue >> 2;
    sensorValue = min(max(sensorValue *
ECG_AMPLIFICATION_FACTOR, 0), 255);

    float x2 = counter * ECG_SPACING;
    float y2 = map(sensorValue, 0, 255, tft.height(), 0);

    if (counter == 0) {
      tft.fillScreen(BLACK);
      tft.setCursor(10, 10);
      tft.setTextColor(WHITE);
      tft.setTextSize(2);
```

```
    tft.print("ECG");
  }

  if (x2 >= 0 && x2 < tft.width()) {
    tft.drawLine(x1, y1, x2, y2, RED);
  }

  x1 = x2;
  y1 = y2;
  counter++;

  if (counter * ECG_SPACING >= tft.width()) {
    counter = 0;
  }

  // R-peak detection and BPM calculation
  if (sensorValue > threshold && !peakDetected) {
    unsigned long currentTime = millis();
    if (lastPeakTime > 0) {
      bpm = 60000 / (currentTime - lastPeakTime);
    }
    lastPeakTime = currentTime;
    peakDetected = true;
  } else if (sensorValue < threshold) {
```

```
      peakDetected = false;
    }

    // Display BPM
    tft.fillRect(10, 40, 150, 30, BLACK); // Clear old BPM
    tft.setCursor(10, 40);
    tft.setTextColor(WHITE);
    tft.setTextSize(2);
    tft.print("BPM: ");
    tft.print(bpm);
  }
}

void drawGrid() {
  for (int i = 0; i < tft.width(); i += 20) {
    tft.drawLine(i, 0, i, tft.height(), GRID_COLOR);
  }
  for (int i = 0; i < tft.height(); i += 20) {
    tft.drawLine(0, i, tft.width(), i, GRID_COLOR);
  }
}

void drawBoldText(const char* text, int x, int y, int size, uint16_t
color) {
```

```
    tft.setTextSize(size);
    for (int dx = -1; dx <= 1; dx++) {
        for (int dy = -1; dy <= 1; dy++) {
            tft.setCursor(x + dx, y + dy);
            tft.setTextColor(color);
            tft.print(text);
        }
    }
    tft.setCursor(x, y);
    tft.setTextColor(color);
    tft.print(text);
}
```